



# MAUI For AsReaderA24D SDK

SDK マニュアル V1.0

## 変更履歴

No.	バージョン	変更内容	日付
1	1.0	新規作成	2024/04/07

# 目次

<b>1 SDK の追加</b> .....	<b>4</b>
1.1 SDK をインポートする.....	4
1.2 権限の追加.....	10
<b>2 SDK の使用</b> .....	<b>11</b>
2.1 SDK のインポート.....	11
2.2 「MyAsReaderDeviceDelegate」デリゲートクラスを作成して、デリゲートメソッドを実現する.....	11
2.3 「BarcodeAsReaderDeviceDelegate」デリゲートクラスを作成して、デリゲートメソッドを実現する.....	12
2.4 AsReader オブジェクトを初期化する.....	12
2.5 OnConnect を実現する.....	12
2.6 OnReadBarcode を実現する.....	13
2.7 Barcode スキャン開始.....	13
2.8 Barcode スキャン停止.....	13
<b>3 AsReader A24D SDK MAUI .Net の変更</b> .....	<b>14</b>
<b>3 Api の使用</b> .....	<b>17</b>



アプリケーションを作成する開発環境は、以下のものを推奨します。

開発環境：

.Net バージョン：.Net 7.0

Visual Studio: Visual Studio Community 2022 for Mac 17.6.7 (build 417)

Android Studio：

Android Studio Giraffe | 2022.3.1 Patch 4

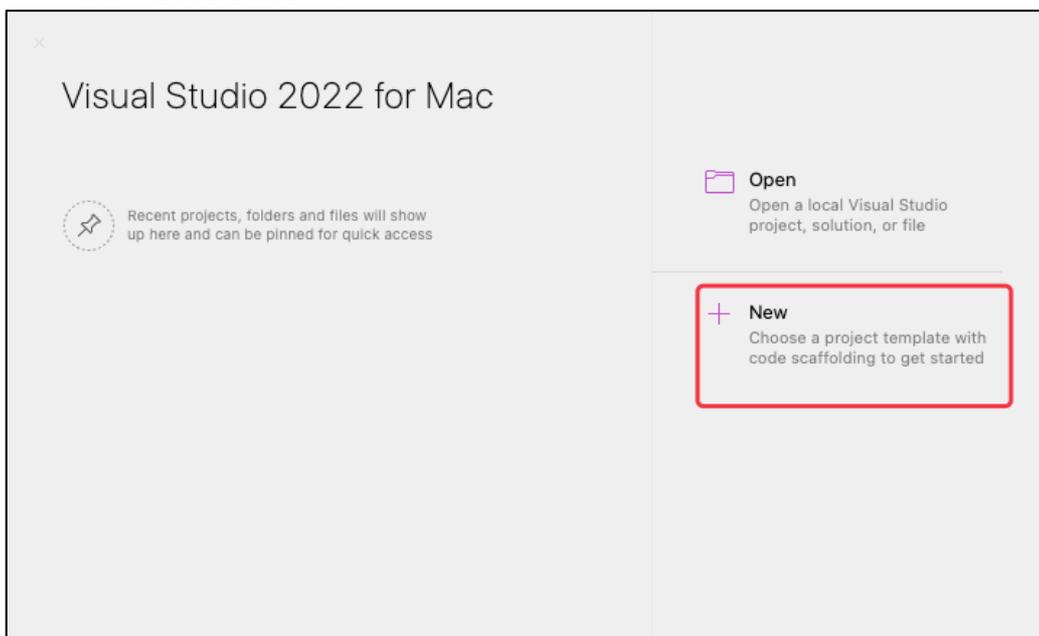
Build #AI-223.8836.35.2231.11090377, built on November 14, 2023

この文書の説明には、上記開発環境を使用しています。

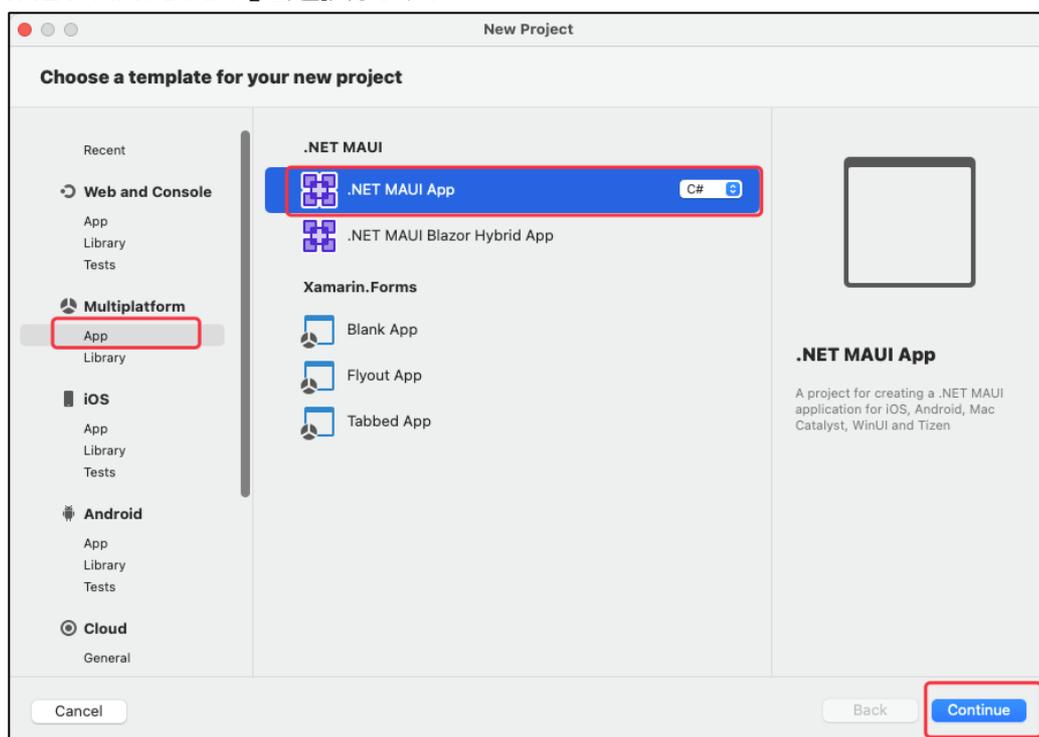
## 1 SDK の追加

### 1.1 SDK をインポートする

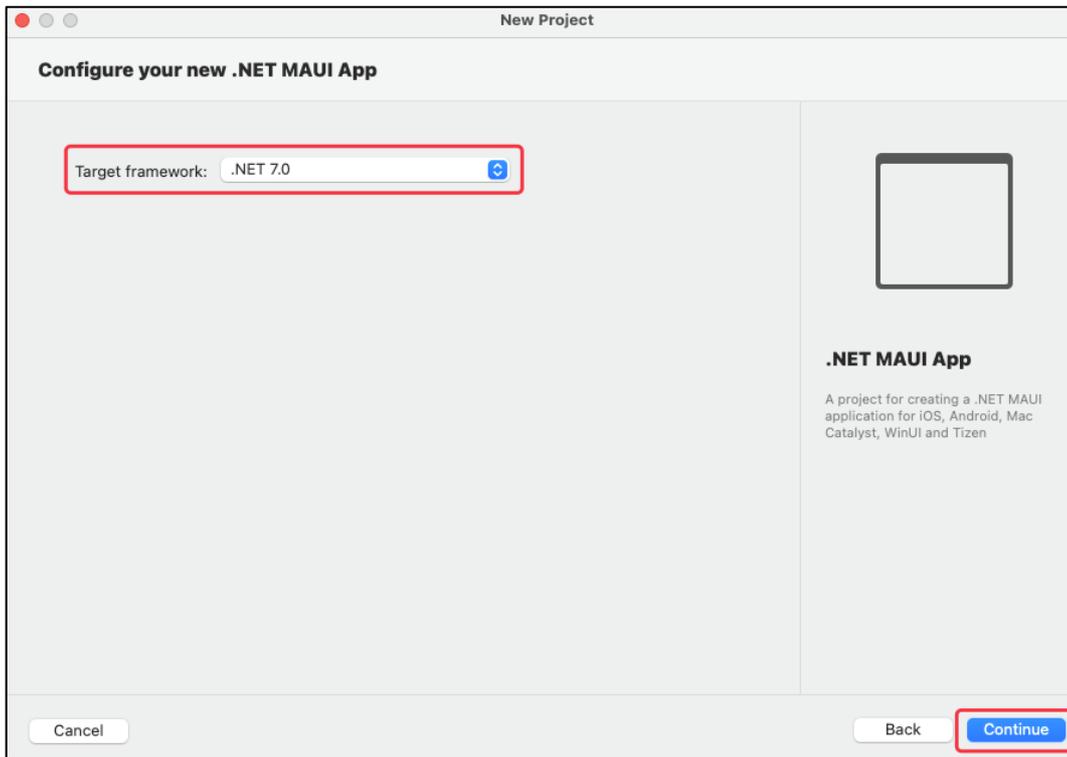
1. AsReaderA24DSDK をローカルにダウンロードします。
2. プロジェクトを新規作成します。



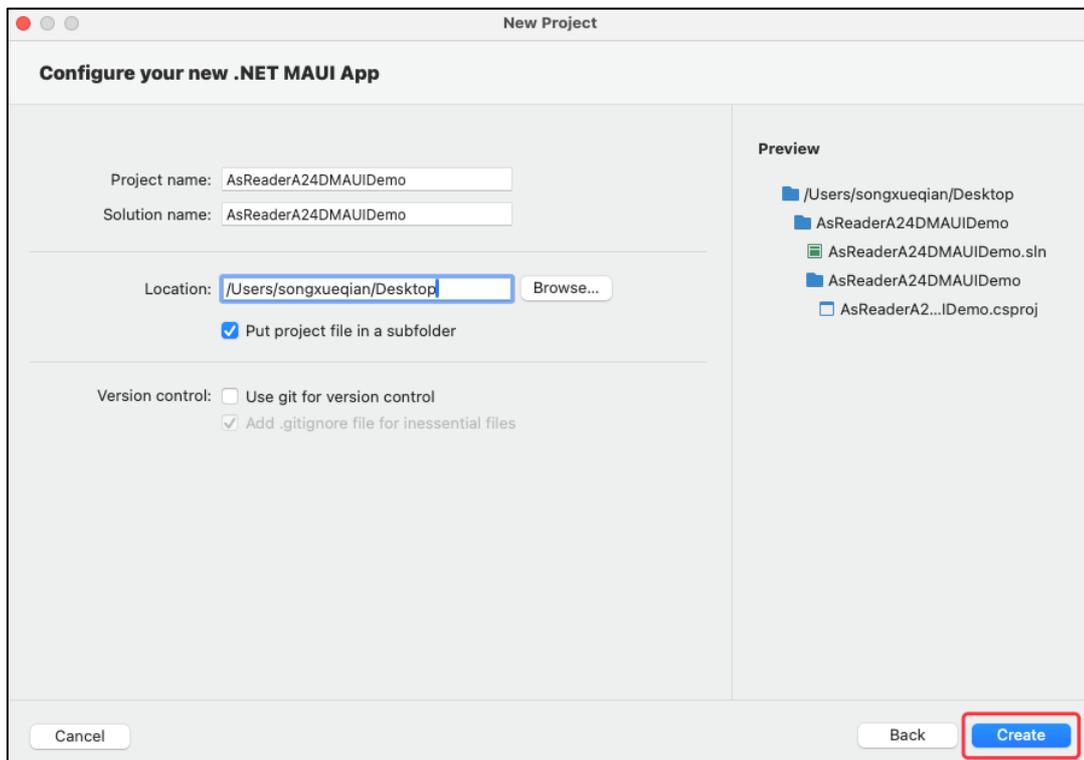
3. 「.NET MAUI APP」を選択します。



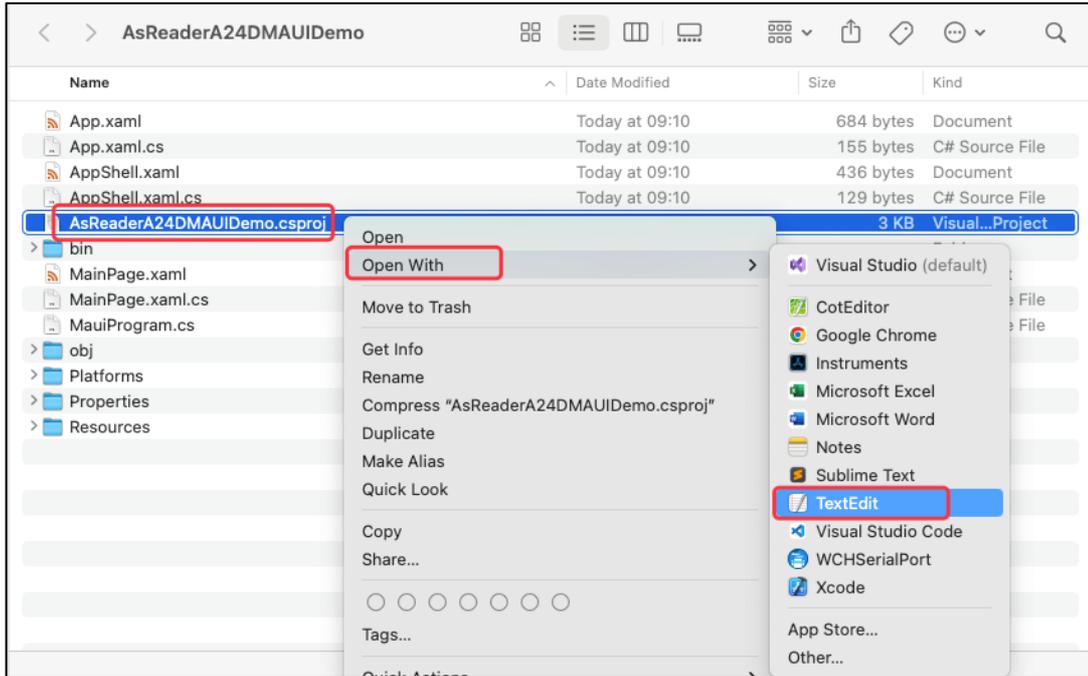
## 4. 「.NET 7.0」を選択します。



## 5. アプリ名とロケーションを作成します。



6. 「AsReaderA24DMAUIDemo.csproj」を右クリックして、TextEdit で開きます。



7. 下図赤枠内（net7.0-android）で、必要としない全てのフレームを削除します。（仅 Android 平台支持，多平台不需要当前操作）

```

<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFrameworks>net7.0-android;net7.0-ios;net7.0-maccatalyst</TargetFrameworks>
    <TargetFrameworks Condition="'$(MSBuild)::IsOSPlatform('windows')'>$(TargetFrameworks);net7.0-
windows10.0.19041.0</TargetFrameworks>
    <!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://
github.com/Samsung/Tizen.NET -->
    <!-- <TargetFrameworks>$(TargetFrameworks);net7.0-tizen</TargetFrameworks> -->
    <OutputType>Exe</OutputType>
    <RootNamespace>AsReaderA24DMAUIDemo</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

    <!-- Display name -->
    <ApplicationTitle>AsReaderA24DMAUIDemo</ApplicationTitle>
  
```

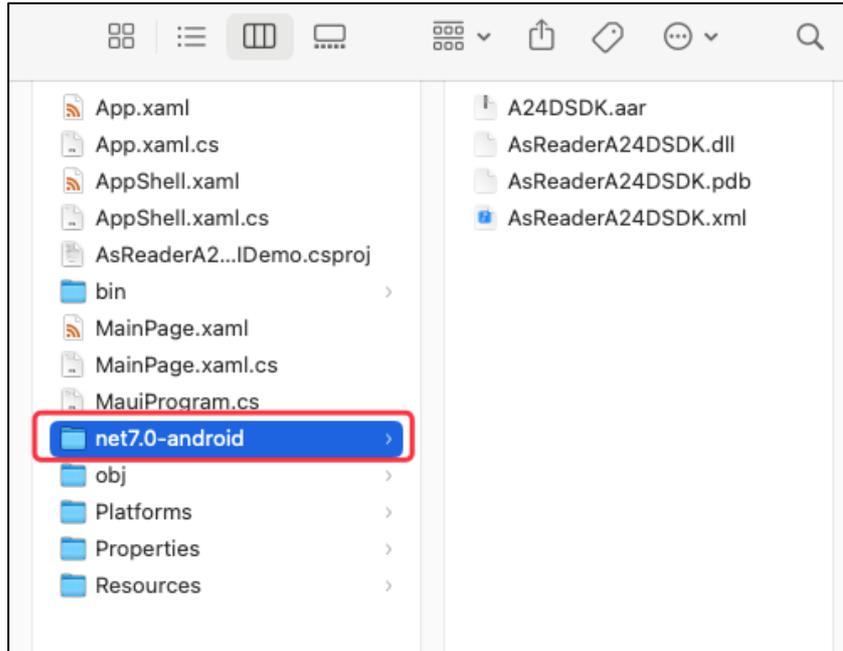
8. 削除完了後は下図の通りです。

```

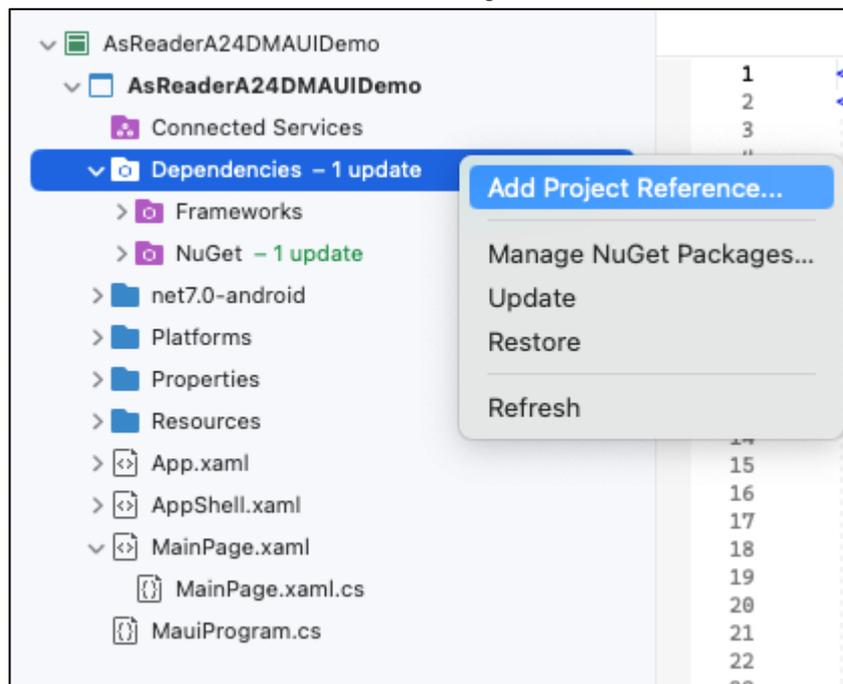
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFrameworks>net7.0-android</TargetFrameworks>
    <TargetFrameworks Condition="'$(MSBuild)::IsOSPlatform('windows')'>$(TargetFrameworks);net7.0-
windows10.0.19041.0</TargetFrameworks>
    <!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://
github.com/Samsung/Tizen.NET -->
    <!-- <TargetFrameworks>$(TargetFrameworks);net7.0-tizen</TargetFrameworks> -->
    <OutputType>Exe</OutputType>
    <RootNamespace>AsReaderA24DMAUIDemo</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

    <!-- Display name -->
    <ApplicationTitle>AsReaderA24DMAUIDemo</ApplicationTitle>
  
```

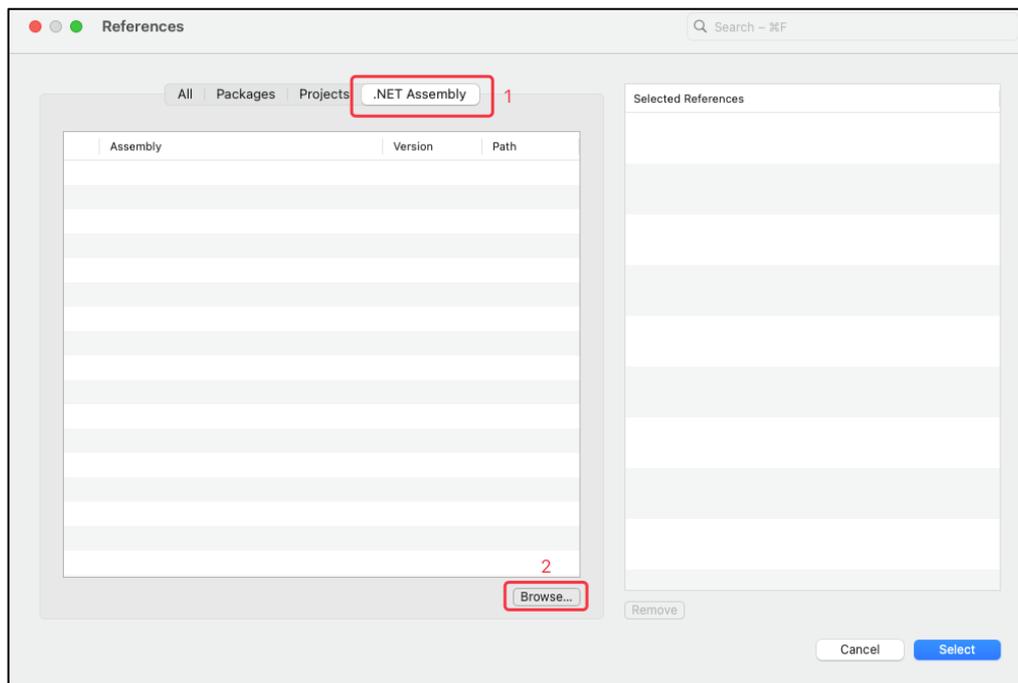
9. ローカルにダウンロードした AsReaderA24DSDK をプロジェクトフォルダーにコピーします。



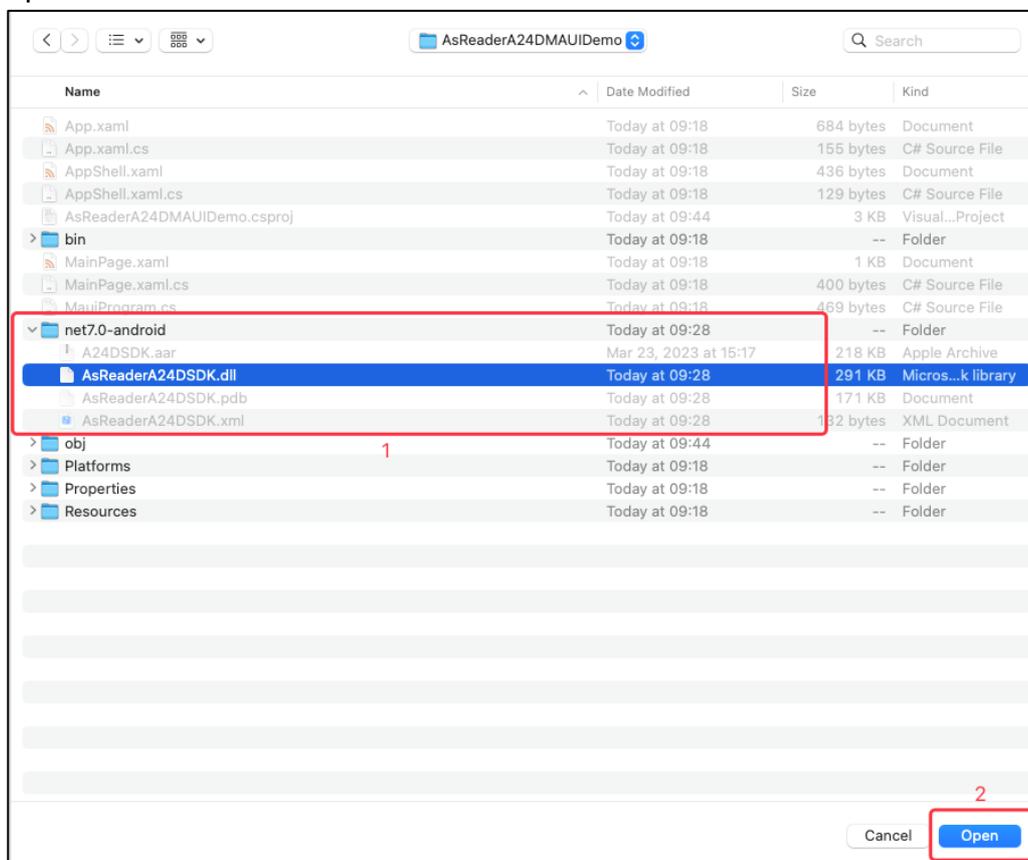
10. Dependencies -> 右クリック -> Add Project Reference...



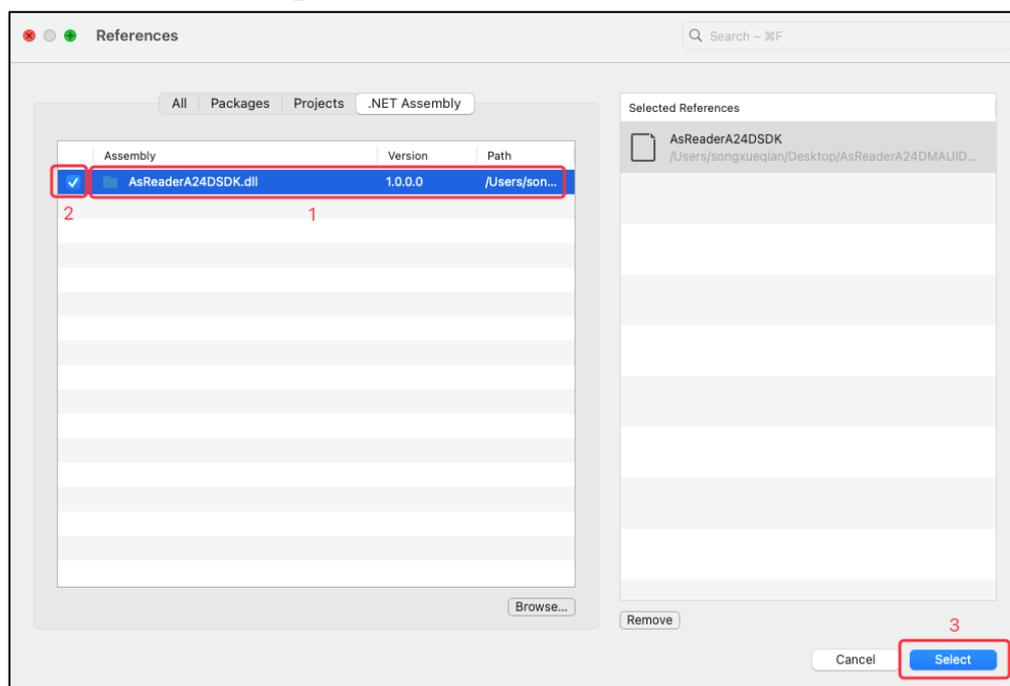
11. ポップアップしたウィンドウに「.Net Assembly」（下図 1）をクリックして、「Browse…」（下図 2）を選択します。



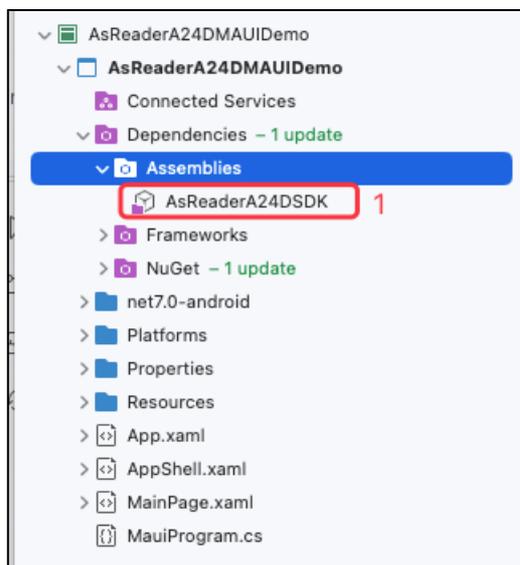
12. ポップアップしたウィンドウに「AsReaderA24DSDK.dll」（下図 1）を選択して、「Open」ボタン（下図 2）をクリックします。



13. 「References」ウィンドウのリストにある「AsReaderA24DSDK.dll」左側のチェックボックスを選択して、「Select」ボタンをクリックします。



14. 追加後、「Assemblies」に「AsReaderA24DSDK」（下図 1）が表示されます。



15. マルチプラットフォームを使用する場合、以下の操作を行ってください。

```
<ItemGroup Condition="'$(TargetFramework)' == 'net7.0-android'">
  <Reference Include="AsReaderA24DSDK">
    <HintPath>net7.0-android\AsReaderA24DSDK.dll</HintPath>
  </Reference>
</ItemGroup>
</Project>
```

## 1.2 権限の追加

AndroidManifest.xml に権限を追加します。

```
AndroidManifest.xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.asreader.a24d.maui.demo" android:versionCode="1" android:versionName="1.0.0">
  <application android:allowBackup="true" android:icon="@mipmap/appicon" android:supportsRtl="true" android:label="AsReaderA24DMAUIDemo"></application>
  <uses-feature android:name="android.hardware.usb.host" />
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-sdk android:minSdkVersion="29" android:targetSdkVersion="33" />
</manifest>
```

## 2 SDK の使用

### 2.1 SDK のインポート

アプリケーションの SDK を使用するクラスに using 文で SDK をインポートします。

```
#if ANDROID
using AsReaderA24DSDK;
#endif
```

### 2.2 「MyAsReaderDeviceDelegate」デリゲートクラスを作成して、デリゲートメソッドを実現する

※Platforms の Android フォルダに作成することをお勧めします。

```
public class MyAsReaderDeviceDelegate : Java.Lang.Object, IDeviceManagerCallback
{
    public MyAsReaderDeviceDelegate()
    {
    }
    public void OnConnect(bool isConnect)
    {
        if (isConnect)
        {
            //Connection successful
        } else
        {
            //Disconnect
        }
    }
    public void OnBatteryStateReceived(int battery)
    {
    }
    public void OnReaderInfoReceived(IDictionary<string, string> info)
    {
    }
    public void OnDeviceManagerError(AsReaderError.ErrorCode errorCode)
    {
    }
    public void OnTriggerEventReceived(DeviceConst.DeviceTriggerEvent triggerEvent)
    {
    }
}
```

## 2.3 「BarcodeAsReaderDeviceDelegate」デリゲートクラスを作成して、デリゲートメソッドを実現する

※Platforms の Android フォルダに作成することをお勧めします。

```
public class BarcodeAsReaderDeviceDelegate : Java.Lang.Object, IBarcodeManagerCallback
{
    public BarcodeAsReaderDeviceDelegate()
    {
    }
    public void OnReceivedBarcodeDecodeData(string decodeData, IDictionary<string, string> parameter)
    {
    }
    public void OnReceivedBarcodeDecodeData(byte[] decodeData)
    {
    }
    public void OnBarcodeManagerError(BarcodeManagerError errorCode)
    {
    }
}
```

## 2.4 AsReader オブジェクトを初期化する

```
MyAsReaderDeviceDelegate myAsReaderDeviceDelegate = new MyAsReaderDeviceDelegate();
AsReader.Instance.DeviceManager.SetCallback(myAsReaderDeviceDelegate);
BarcodeAsReaderDeviceDelegate barcodeAsReaderDeviceDelegate = new BarcodeAsReaderDeviceDelegate();
AsReader.Instance.BarcodeManager.SetCallback(barcodeAsReaderDeviceDelegate);
AsReader.Instance.DeviceManager.SetPackageName(MainActivity.instance.PackageName);
AsReader.Instance.Initialize(MainActivity.instance, ConnectionType.Usb);
```

## 2.5 OnConnect を実現する

```
public void OnConnect(bool isConnect)
{
    if (isConnect)
    {
        //Connection successful
    } else
    {
        //Disconnect
    }
}
```

## 2.6 OnReadBarcode を実現する

```
public void OnReceivedBarcodeDecodeData(string decodeData, IDictionary<string, string> parameter)
{
    ...
}
public void OnReceivedBarcodeDecodeData(byte[] decodeData)
{
    ...
}
```

## 2.7 Barcode スキャン開始

```
AsReader.Instance.BarcodeManager.StartDecode();
```

## 2.8 Barcode スキャン停止

```
AsReader.Instance.BarcodeManager.StopDecode();
```

## 3 AsReader A24D SDK MAUI .Net の変更

デモアプリはビルドした時に使用したバージョンのツールでしか使えません。

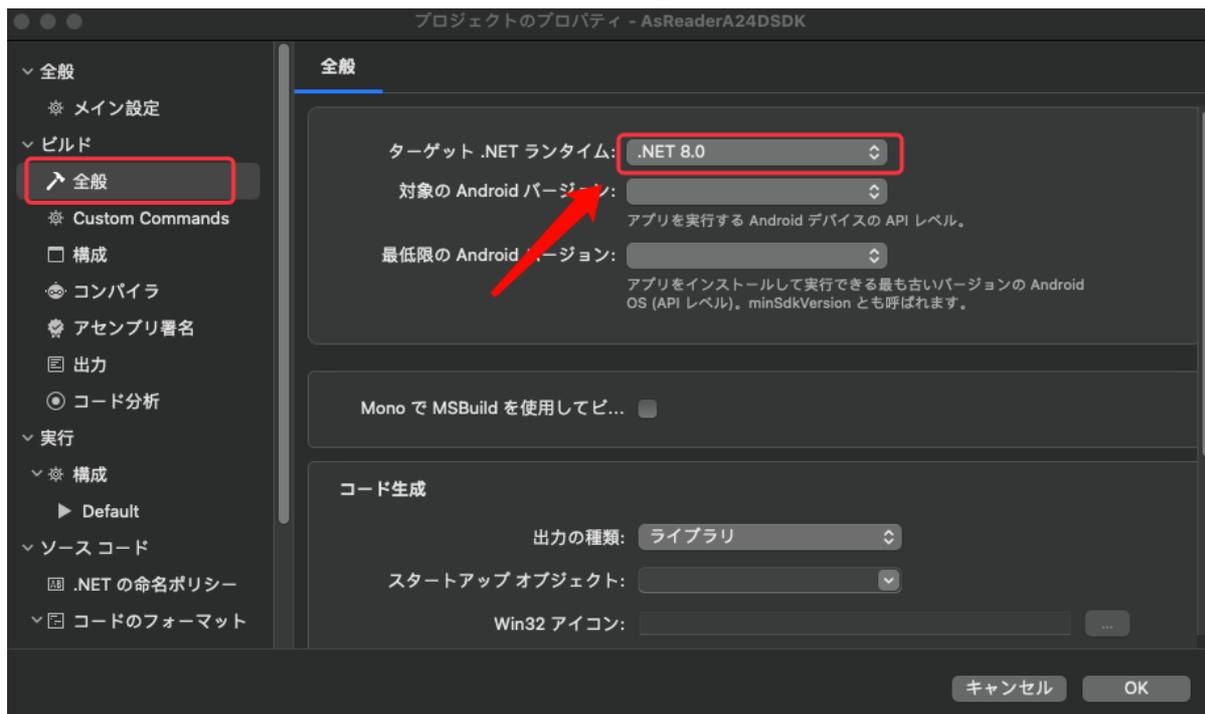
例：「.net MAUI 8」でビルドした場合、「.net MAUI 8」だけで使えます。「.net MAUI 7」や「.net MAUI 9」で使えません。

以下は例として「.net MAUI 7」を「.net MAUI 8」に変更する方法です。

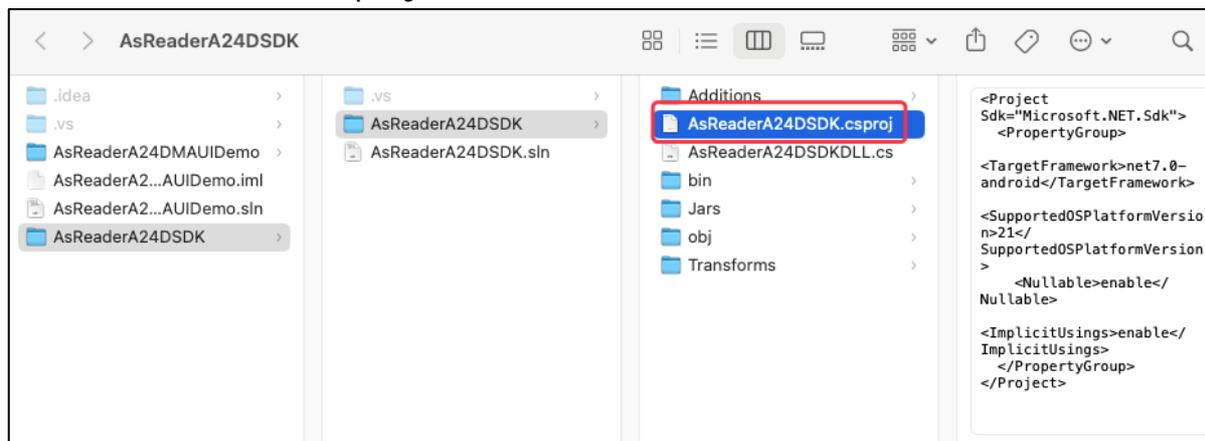
### 1. プロジェクトを右クリック->プロパティ



2. 全般->ターゲット.NET ランタイムに「.NET 8.0」を選択します。



3. プロジェクトフォルダに.csproj を選択して、editor で開きます。



## 4. 以下の修正を行います。



```
@@ -1,8 +1,11 @@
<Project Sdk="Microsoft.NET.Sdk">
- <PropertyGroup>
-   <TargetFramework>net7.0-android</TargetFramework>
-   <SupportedOSPlatformVersion>21</SupportedOSPlatformVersion>
-   <Nullable>enable</Nullable>
-   <ImplicitUsings>enable</ImplicitUsings>
- </PropertyGroup>
</Project>
No newline at end of file

@@ -1,8 +1,11 @@
1 <Project Sdk="Microsoft.NET.Sdk">
2 <PropertyGroup>
3 <TargetFramework>net8.0-android</TargetFramework>
4 <SupportedOSPlatformVersion>21</SupportedOSPlatformVersion>
5 <Nullable>enable</Nullable>
6 <ImplicitUsings>enable</ImplicitUsings>
7 <UseMaui>true</UseMaui>
8 <MauiVersion>8.0.3</MauiVersion>
9 <SkipValidateMauiImplicitPackageReferences>true</SkipValidateMauiImplicitPackageReferences>
10 </PropertyGroup>
11 </Project>
No newline at end of file
```

<TargetFramework>net7.0-android</TargetFramework>

↓

<TargetFramework>net8.0-android</TargetFramework>

以下の配置を追加します。

<UseMaui>true</UseMaui>

<MauiVersion>8.0.3</MauiVersion>

<SkipValidateMauiImplicitPackageReferences>true</SkipValidateMauiImplicitPackageReferences>

修正後にファイルを保存します。

4.保存後に改めてプロジェクトをローディングします。



## 3 Api の使用

Api の使用について、Java の SDK マニュアルを参照してください。  
「ASR-A24D\_JavaDoc\_Android\_JP\_1\_0」